

Eclipse Modeling Project and OMG™ Standards

Version 1.0, September 25, 2006

Richard C. Gronback, Borland Software Corporation, Connecticut, USA – richard.gronback@borland.com

Abstract

A number of questions arise from the Eclipse Modeling Project's charter related to the need to "foster a good working relationship with external [standards] organizations." The extent of such relationships has yet to be discussed. Does it make sense for the Eclipse Modeling Project to produce true reference implementations for OMG modeling language specifications, and if so, what relationship between the organizations would be appropriate? If it does not make sense, what is the benefit of the goal, a "good working relationship," to either organization or to consumers of these technologies?

1. Introduction

As stated in its charter, *"the importance of supporting industry standards is critical to the success of the Modeling project, and to Eclipse in general. The role of the Modeling project in the support of industry standards is to enable their creation and maintenance within the Eclipse community. Furthermore, as standards bodies such as the Object Management Group (OMG) have a strong modeling focus, the Modeling project needs to facilitate communication and outreach through its PMC and project contributors to foster a good working relationship with external organizations."*

While this paper discusses OMG [1] standards implemented within the Eclipse Modeling Project [2], the discussion is also pertinent to interaction with other standards organizations.

It should be noted that no formal communication has yet taken place between the OMG and leadership of the Eclipse Modeling Project regarding this aspect of the charter. Indeed, there is no definition of what "formal communications" would be. The OMG has a "Liaison Committee" and its board authorizes the OMG president some independence in managing relationships with other organizations. Both the outline and the details of how a relationship will be created and sustained are yet to be determined.

This situation raises a number of questions, which are the focus of this paper. It may turn out that the relationship should remain very informal, with no explicit commitment or expectation that implementations found in the Modeling project represent so-called "reference implementations" of OMG standards, as described in [3].

2. Standards

A number of standards are already implemented, or plan to be implemented as part of the new Eclipse Modeling Project. Below is a list of those related to the OMG. Note that other projects at Eclipse also implement OMG standards, e.g. the Software Process Engineering Model (SPEM) [4] as part of the Eclipse Process Framework (EPF) [5] project.

UML2 [6] is currently implemented as part of the Eclipse UML2 project [7]. This project has existed for some time at Eclipse, and will provide diagramming capabilities as part of the expansion of this project into the new Model Development Tools (MDT) project [8].

OCL [9] is currently implemented as part of the EMF Technology project [10] and will also move to MDT.

MOF™ [11] (EMOF) is closely aligned with the Eclipse Modeling Framework (EMF) [12] Ecore metamodel, and will form the basis of most Modeling project technologies. The topic of aligning these two has been ongoing.

XMI [13] is supported by EMF and is used to by the UML2 project and others.

MOF™ Query / Views / Transformations [14] is part of the Model-to-Model Transformation (M2M) project [15], which plans to begin by implementing its Core language.

MOF™ Model to Text Transformations (adopted at a recent OMG meeting) ad/2006-09-03 is likely to be implemented within the context of the M2M project as well.

UML Diagram Interchange [16] is not currently implemented, but has prompted many questions from the community regarding its implementation, particularly with the introduction of UML diagramming planned for MDT.

UML Human-Usable Textual Notation [17] is also not currently implemented, but would fall into the placeholder Textual Modeling Framework (TMF) project within Modeling. There has been some interest in providing general tooling for supporting textual concrete syntaxes for modeling languages.

Business Process Modeling Notation [18] is currently within the scope of the SOA Tools Project [19] at Eclipse, but also seems appropriate to the MDT project. In this case, the former would focus on BPEL generation using the defined mapping, while the latter would focus on general-purpose business process modeling using the standard notation and domain metamodel.

3. Reference Implementations

One of the most important aspects of a good working relationship between a standards provider and an implementation-focused organization is the promise of providing a reference implementation for published specifications. In the past, specifications such as the UML have suffered from interoperability issues among vendors who had different interpretations or implementation goals. The introduction of XMI, well-defined compliance levels, Diagram Interchange specification, etc. were intended to improve the situation, but have largely failed to deliver and now compound the problem. By developing a reference implementation in parallel with the

specification, ambiguities and defects can be minimized earlier and serve the larger community through delivery of a platform upon which to implement commercial products.

Although Eclipse is seen as providing de facto standards in some areas, is it proper for the OMG and Eclipse to move toward an agreement by which Eclipse implementations of OMG standards are considered the true reference implementations, although not in an exclusive manner? With that, what additional restrictions, agreements, policies, etc. regarding contribution from interested members of both will need to be added in order for this to occur? What level of compliance and quality is sufficient to satisfy the Eclipse requirement of "high quality frameworks and exemplary, extensible tools" but where commercial vendors will be able to provide value-add functionality for profitability?

4. Membership

Currently, the Eclipse Foundation is not a member of the OMG, and the OMG is not a member of the Eclipse Foundation. The prospect of each organization becoming a member of the other is under investigation, but raises a question of what level of interaction and commitment this brings, particularly as corporate members of each are often involved in, and provide contributions to, both of these organizations.

What are the best techniques for aligning standards organization activities with reference implementation project team activities? Should members be required to participate in both contribution areas, where applicable? What does it mean for the Eclipse Foundation to be a member of the OMG, and vice versa? What role would the Foundation representative have within the context of the OMG, and how would they coordinate with fellow members from the Eclipse community? What if there are competing goals among members? Are there new working models that would be more productive, and perhaps never before explored in this context?

5. Specification Delivery

Specifications with defined metadata should be delivered in a serialized format, preferably XML. This is required by the standard RFP template for new specifications, but has not been mandated or required for all specifications currently published by the OMG. The BPMN specification, for example, has no published format for its described domain model. Only natural language and a table in Annex B provide the definition, which fails to leverage MOF as its metamodel, or OCL for defining its constraints.

Graphical notations (concrete syntax) are typically provided by drawings and natural language descriptions. While these are typically sufficient for describing the elements, they are not as precise as they could be and must be recreated in order to utilize them in modeling tools.

The delivery of specifications in formats that are machine consumable, particularly if used as inputs to generative tooling frameworks, should be an obvious benefit to those involved in specification, implementation, and consumption of these

technologies.

Should it be a goal for Eclipse-based tooling to serve as the standard environment for the design and serialization of MOF-based models using XML for publication in specifications? Should graphical notations be delivered in a standard format, such as SVG?

6. Domain vs. Notation vs. Mapping

The UML specification contains domain (abstract) syntax and semantics, OCL constraints, and graphical (concrete) syntax, accompanied by natural language description and mapping to the domain.

In the design and implementation of the Eclipse Graphical Modeling Framework (GMF) [20], it was decided that graphical notation should be defined in a model separate from the domain, with its mapping relationship defined in yet another model. The benefits of this approach are that it separates notation for potential reuse in other domains (i.e. if you consider the popularity of the UML Class diagram elements in domains other than UML), and allows for the mapping of notation elements to domain elements to be rigorously defined. In this case, metamodels are provided for the graphical definition, the mapping definition, and a tooling definition. In the runtime component, a notation model is used to persist diagram information, and is similar in concept to the Diagram Interchange (DI) specification [16]. Knowledge of the runtime notation model is only required during the creation of a generator model from the mapping model.

It would seem reasonable for specifications to be delivered in a similar manner, where the abstract model is described separately from the concrete syntax, and where they are only related using a mapping definition. This approach will provide proper separation of concerns, and allow for the generation of graphical editors for various domain models.

Does it make sense for specifications with a graphical concrete syntax to provide a formal mapping definition between its graphical and domain models, and also to their serialization model? And, should graphical notations be defined in term of a graphical definition metamodel, SVG, or another standard?

7. Specification Compliance

There are generally provided a set of conformance criteria to be met when implementing a specification. With improved collaboration between implementation and specification organizations, it can be expected that some level of minimum compliance level be achieved, so as to provide a proper reference implementation. There are cases today where implementations at Eclipse are well aligned, or nearly aligned with OMG specifications. For example, the UML2 project strives to provide a compliant implementation of the UML2 metamodel using the nearly EMOF-compliant Eclipse EMF project.

Should implementations be required to provide the highest level of compliance to defined specification acceptance criteria? Or, is

a "best effort" approach adequate? What actions can or should be taken to provide specification alignment and/or conformance?

8. Implementations Influencing Specifications

There may be the case where existing implementations are close to a specification, yet not fully compliant. For example, the EMF project's Ecore model is not quite aligned with the EMOF specification. The complication here is that EMF has a significant client base and must follow the Quality API and migration guidelines provided by the Eclipse Foundation for its projects.

Another case of an existing implementation impacting a specification lies with the UML Diagram Interchange (DI) specification and the GMF runtime notation model. A brief discussion of how the latter could be used to impact the former had started, but has not yet led to a working relationship to achieve this goal.

In the case of an existing implementation which is closely aligned with a specification and whose list of clients is significant, how much influence can its implementation have on the specification? Does it make sense to force the migration of so many clients away from something popular and working in order to achieve compliance with a specification?

9. Open and Transparent Nature

In the case of Eclipse, contributions are done in the open, with an emphasis on meritocracy as the basis for achieving more responsibilities within the community. Transparency is essential to the open source process at Eclipse, yet is somewhat different from the specification development process at the OMG [21]. Perhaps this is an area where the two organizations can influence one another?

If the development of a reference implementation is done in the open, it would follow that the developing version of the specification itself must be available. Otherwise, there would need to be a serial process of first developing the specification, publishing, and then implementing it, which eliminates the benefits of validating the specification while developing an implementation in parallel.

Can the process of developing standards be done in a more open and transparent manner, with an emphasis on addressing the needs of a developing reference implementation? Alternatively, could Eclipse support a model where source is not open until it reaches a required level of alignment with ongoing specification work?

10. Conclusion

In summary, there are many challenges facing the statement in the Modeling project's charter related to its future relationship with standards bodies such as the OMG. A relationship that is too informal will be unlikely to yield the desired results, while a relationship that is strictly defined and enforced will likely limit the progress of implementation. The right balance will clearly benefit both of these organizations, their members, and ultimately the customers of commercial products that are standards-based.

It remains a task of the Modeling project leadership to contact and begin discussion with the OMG and other standards bodies to explore the process and working model for this mutually beneficial relationship.

Acknowledgements

Thanks to Karl Frank, Tom Gullion, and Ed Merks for their review of this paper and valuable input.

References

1. Object Management Group (OMG)
<http://www.omg.org>
2. Eclipse Modeling Project (EMP)
<http://www.eclipse.org/modeling>
3. OMG Specification and Products
<http://www.omg.org/gettingstarted/specsandprods.htm#SpecProd>
4. Software Process Engineering Metamodel (SPEM) specification
http://www.omg.org/technology/documents/modeling_spec_catalog.htm#SPEM
5. Eclipse Process Framework (EPF)
<http://www.eclipse.org/epf/>
6. Unified Modeling Language™ (UML®) specification
http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML
7. Eclipse UML2 Project <http://www.eclipse.org/uml2>
8. Eclipse Model Development Tools (MDT) Project
<http://www.eclipse.org/proposals/mdt>
9. Object Constraint Language (OCL) specification
http://www.omg.org/technology/documents/modeling_spec_catalog.htm#OCL
10. EMF Technology OCL Project
<http://www.eclipse.org/emft/projects/ocl#ocl>
11. Meta-Object Facility (MOF™) specification
http://www.omg.org/technology/documents/modeling_spec_catalog.htm#MOF
12. Eclipse Modeling Framework (EMF) Project
<http://www.eclipse.org/emf>
13. XML Metadata Interchange (XMI) specification
http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XMI
14. MOF™ Query / View / Transformation (QVT) specification

- http://www.omg.org/technology/documents/modeling_spec_catalog.htm#MOF_QVT
15. Eclipse Model-to-Model Transformation (M2M) Project <http://www.eclipse.org/proposals/m2m>
 16. UML Diagram Interchange (DI) specification http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML_DI
 17. UML Human-Usable Textual Notation (HUTN) specification http://www.omg.org/technology/documents/modeling_spec_catalog.htm#HUTN
 18. Business Process Modeling Notation (BPMN) specification http://www.omg.org/technology/documents/bms_spec_catalog.htm#BPMN
 19. Eclipse SOA Tools Project (STP) <http://www.eclipse.org/stp>
 20. Eclipse Graphical Modeling Framework (GMF) Project <http://www.eclipse.org/gmf>
 21. OMG Technology Adoption Process <http://www.omg.org/gettingstarted/processintro.htm>